Int J Comput Vis (2014) 107:191–202
DOI 10.1007/s11263-013-0683-3

Max-Margin Early Event Detectors

Minh Hoai · Fernando De la Torre

Received: 30 December 2012 / Accepted: 27 November 2013 / Published online: 13 December 2013

Abstract The need for early detection of temporal events from sequential data arises in a wide spectrum of applications ranging from human-robot interaction to video security. While temporal event detection has been extensively studied, early detection is a relatively unexplored problem. This paper proposes a maximum-margin framework for training temporal event detectors to recognize partial events, enabling early detection. Our method is based on Structured Output SVM, but extends it to accommodate sequential data. Experiments on datasets of varying complexity, for detecting facial expressions, hand gestures, and human activities, demonstrate the benefits of our approach.

Keywords Early detection · Event detection · Structured output learning

1 Introduction

The ability to make reliable early detection of temporal events has many potential applications in a wide range of fields, including security (e.g., pandemic attack detection), environmental science (e.g., tsunami warning), healthcare (e.g., risk-of-falling detection), entertainment (e.g., gaming), and robotics (e.g., affective computing). A temporal event has a duration, and by early detection, we mean to detect the event as soon as possible, *after it starts but before it ends*, as illustrated in Fig. 1. To see why it is important to detect events before they finish, consider a concrete example of building a robot that can affectively interact with humans. Arguably, a key requirement for such a robot is its ability to accurately and rapidly detect a human's emotional states from facial expression so that appropriate responses can be made in a timely manner. More often than not, a socially acceptable response is to imitate the human's current behavior. This requires facial events such as smiling or frowning to be detected even before they are complete; otherwise, the imitation response would be out of synchronization.

Despite the importance of early detection, few machine learning formulations have been explicitly developed for early detection. Most existing methods for event detection and modeling (e.g., Ke et al. 2005; Smith et al. 2005; Gorelick et al. 2007; Oh et al. 2008; Satkin and Hebert 2010; Klaser et al. 2010; Ali and Shah 2010; Niebles et al. 2010; Shi et al. 2010; Pei et al. 2011; Liu et al. 2011; Marin-Jiménez et al. 2011; Lan et al. 2011; Hoai and De la Torre 2012b; Amer et al. 2012; Yang and Shah 2012) are designed for offline processing. They have a limitation for processing sequential data as they are only trained to detect complete events. But for early detection, it is necessary to recognize partial events, which are ignored in the training process of existing event detectors.

This paper proposes Max-Margin Early Event Detectors (MMED), a novel formulation for training event detectors that recognize partial events, enabling early detection. MMED is based on structured output SVM (SOSVM) (Taskar et al. 2003; Tsochantaridis et al. 2005), but extends it to accommodate the nature of sequential data. In particular, we simulate the sequential frame-by-frame data arrival for training time series and learn an event detector that correctly classifies partially observed sequences. Figure 2 illustrates the key idea behind MMED: partial events are simulated and used as positive training examples. It is important to emphasize that we train a *single* event detector to recognize *all*

M. Hoai (✉) · F. De la Torre
Robotics Institute, Carnegie Mellon University,
5000 Forbes Ave., Pittsburgh, PA15213, USA
e-mail: minhhoai@robots.ox.ac.uk; minhhoai@cmu.edu

F. De la Torre
e-mail: ftorre@cs.cmu.edu

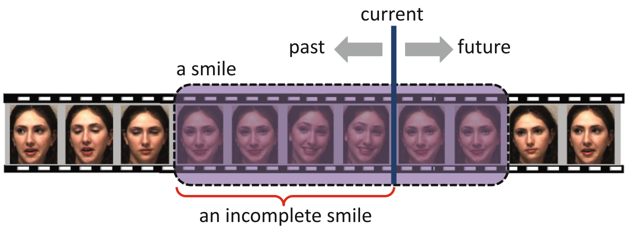

Fig. 1 How many frames do we need to detect a smile reliably? Can we even detect a smile before it finishes? Existing event detectors are trained to recognize complete events only; they require seeing the entire event for a reliable decision, preventing early detection. We propose a learning formulation to recognize partial events, enabling early detection

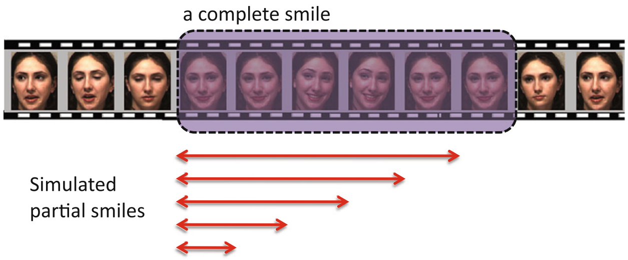

Fig. 2 Given a training time series that contains a complete event, we simulate the sequential arrival of training data and use partial events as positive training examples. The *red* segments indicate the temporal extents of the partial events. We train a *single* event detector to recognize *all* partial events, but our method does more than augment the set of training examples (Color figure online)

partial events. But MMED does more than augment the set of training examples; it trains a detector to localize the temporal extent of a target event, even when the target event has not yet finished. This requires monotonicity of the detection function with respect to the inclusion relationship between partial events—the detection score (confidence) of a partial event cannot exceed the score of an encompassing partial event. MMED provides a principled mechanism to achieve this monotonicity, which cannot be assured by a naive solution that simply augments the set of training examples.

The learning formulation of MMED is a constrained quadratic optimization problem. This formulation is theoretically justified. In Sect. 3.3, we discuss two ways for quantifying the loss for continuous detection on sequential data. We prove that, in both cases, the objective of the learning formulation is to minimize an upper bound of the true loss on the training data.

MMED has numerous benefits. First, MMED inherits the advantages of SOSVM, including its convex learning formulation and its ability to accurately localize event boundaries. Second, MMED, specifically designed for early detection, is superior to SOSVM and other competing methods regarding the timeliness of the detection. Experiments on datasets of varying complexity, including sign language, facial expression, and human actions, showed that our method often made earlier detections while maintaining comparable or even better accuracy.

2 Previous Work

This section discusses previous work on event detection and early detection.

2.1 Event Detection in Computer Vision

Events are integral parts of video, and many techniques for event modeling and detection can be found in the literature of video analysis, including facial expression recognition (e.g., Cohn et al. 2009; Nguyen et al. 2010; Lucey et al. 2010), gesture and sign language interpretation (e.g., Nam et al. 1999; Cooper and Bowden 2009), human action classification (e.g., Bobick and Davis 2001; Efros et al. 2003; Parameswaran and Chellappa 2006; Jhuang et al. 2007; Duchenne et al. 2009; Patron-Perez et al. 2010; Hoai et al. 2011; Reddy and Shah 2012), and activity recognition (e.g., Brand et al. 1997; Yacoob and Black 1999; Chomat and Crowley 1999; Dollár et al. 2005; Tran and Davis 2008; Nguyen et al. 2009; Ryoo and Aggarwal 2009; Brendel and Todorovic 2011). Some of the aforementioned techniques can be used for event detection while the others are only suitable for classification. These techniques, however, are designed for offline processing.

2.2 Early Detection

While event detection has been studied extensively in the literature of computer vision, little attention has been paid to early detection. Davis and Tyagi (2006) addressed rapid recognition of human actions using the probability ratio test. This is a passive method for early detection; it assumes that a generative HMM for an event class, trained in a standard way, can also generate partial events. Schindler and Van Gool (2008), Mauthner et al. (2009), Masood et al. (2011) proposed to train one-frame or two-frame detectors, but these detectors yield high false positive rates (FPR) for many types of events. Ryoo (2011) considered the unseen part of an event as a latent variable and developed two variants of the bag-of-words representation. His approach failed to account for the sequential nature of temporal events, and it mainly addressed the computational issues, not the timeliness or accuracy, of the detection process. Around the same time that our method was being developed (Hoai and De la Torre 2012a), Nowozin and Shotton (2012) proposed low-latency HMMs, but their approach required additional annotation for Kinect data. More recently, Ellis et al. (2013) explored the trade-off between accuracy and observational latency by seeking distinctive canonical human poses.

Previous work on early detection exists in other fields, but its applicability to computer vision is unclear. Neill et al. (2006) studied disease outbreak detection. Their approach, like online change-point detection (Desobry et al. 2005), is based on detecting the locations where abrupt statistical changes occur. This technique, however, cannot be applied to detect temporal events such as smiling and frowning, which must and can be detected and recognized independently of the background. Brown et al. (1992) used the n-gram model for predictive typing, i.e., predicting the next word from previous words. However, it is hard to apply their method to computer vision, which does not have a well-defined language model yet. Early detection has also been studied in the context of spam filtering, where immediate and irreversible decisions must be made whenever an email arrives. Assuming spam messages were similar to one another, Haider et al. (2007) developed a method for detecting batches of spam messages based on clustering. But visual events such as smiling or frowning cannot be detected and recognized just by observing the similarity between constituent frames, because this characteristic is neither requisite nor exclusive to these events.

It is important to distinguish between forecasting and detection. Forecasting predicts the future while detection interprets the present. For example, financial forecasting (e.g., Kim 2003) predicts the next day's stock index based on the current and past observations. This technique cannot be directly used for early event detection because it predicts the raw value of the next observation instead of recognizing the event class of the current and past observations. Perhaps, forecasting the future is a good first step for recognizing the present, but this two-stage approach has a disadvantage because the former may be harder than the latter. For example, it is probably easier to recognize a partial smile than to predict when it will end or how it will progress.

2.3 Learning Formulations for Event Detectors

This section reviews SVM, HMM, and SOSVM, which are among the most popular algorithms for training event detectors. None of them are specifically designed for early detection.

Let $(\mathbf{X}^1, \mathbf{y}^1), \cdots, (\mathbf{X}^n, \mathbf{y}^n)$ be the set of training time series and their associated ground truth annotations for the events of interest. Here we assume each training sequence contains at most one event of interest, as a training sequence containing several events can always be divided into smaller subsequences of single events. Thus, $\mathbf{y}^i = [s^i, e^i]$ consists of two numbers indicating the start and the end of the event in time series $\mathbf{X}^i$. Suppose the length of an event is bounded by l_{min} and l_{max} and we denote $\mathcal{Y}(t)$ the set of length-bounded time intervals from the 1st to the tth frame:

$$\mathcal{Y}(t) = \{\mathbf{y} \in \mathbb{N}^2 | \mathbf{y} \subset [1, t], l_{min} \leq |\mathbf{y}| \leq l_{max}\} \cup \{\emptyset\}.$$

Here $|\cdot|$ is the length function. For a time series $\mathbf{X}$ of length l, $\mathcal{Y}(l)$ is the set of all possible locations of an event; the empty segment, $\mathbf{y} = \emptyset$, indicates no event occurrence. For an interval $\mathbf{y} = [s, e] \in \mathcal{Y}(l)$, let $\mathbf{X_y}$ denote the subsegment of $\mathbf{X}$ from frame s to e inclusive. Let $g(\mathbf{X})$ denote the output of the detector, which is the segment that maximizes the detection score:

$$g(\mathbf{X}) = \underset{\mathbf{y} \in \mathcal{Y}(l)}{\operatorname{argmax}} f(\mathbf{X_y}; \theta). \tag{1}$$

The output of the detector may be the empty segment, and if it is, we report no detection. $f(\mathbf{X_y}; \theta)$ is the detection score of segment $\mathbf{X_y}$, and θ is the parameter of the score function. Note that the detector searches over temporal scales from l_{min} to l_{max}. In testing, this process can be repeated to detect multiple target events, if more than one event occurs.

How is θ learned? Binary SVM methods learn θ by requiring the score of positive training examples to be greater than or equal to 1, i.e., $f(\mathbf{X}^i_{\mathbf{y}^i}; \theta) \geq 1$, while constraining the score of negative training examples to be smaller than or equal to -1. Negative examples can be selected in many ways; a simple approach is to choose random segments of the training time series that do not overlap with positive examples. HMM methods define $f(\cdot, \theta)$ as the log-likelihood and learn θ that maximizes the total log-likelihood of positive training examples, i.e., maximizing $\sum_i f(\mathbf{X}^i_{\mathbf{y}^i}; \theta)$. HMM methods ignore negative training examples. SOSVM methods learn θ by requiring the score of a positive training example $\mathbf{X}^i_{\mathbf{y}^i}$ to be greater than the score of any other segment from the same time series, i.e., $f(\mathbf{X}^i_{\mathbf{y}^i}; \theta) > f(\mathbf{X}^i_{\mathbf{y}}; \theta)\ \forall \mathbf{y} \neq \mathbf{y}^i$. SOSVM further requires this constraint to be well satisfied by a margin: $f(\mathbf{X}^i_{\mathbf{y}^i}; \theta) \geq f(\mathbf{X}^i_{\mathbf{y}}; \theta) + \Delta(\mathbf{y}^i, \mathbf{y})\ \forall \mathbf{y} \neq \mathbf{y}^i$, where $\Delta(\mathbf{y}^i, \mathbf{y})$ is the loss of the detector for outputting $\mathbf{y}$ when the desired output is $\mathbf{y}^i$ (Nguyen et al. 2010). Though optimizing different learning objectives and constraints, all of these aforementioned methods use the same set of positive examples. They are trained to recognize *complete* events only, inadequately prepared for the task of early detection.

3 Max-Margin Early Event Detectors

As explained above, existing methods do not train detectors to recognize partial events. Consequently, using these methods for online prediction would lead to unreliable decisions as we will illustrate in the experimental section. This section derives a learning formulation to address this problem. We use the same notation as described in Sect. 2.3.

3.1 Learning with Simulated Sequential Data

Let $\varphi(\mathbf{X_y})$ be the feature vector for segment $\mathbf{X_y}$. We consider a linear detection score function:

$$f(\mathbf{X_y};\theta) = \begin{cases} \mathbf{w}^T\varphi(\mathbf{X_y}) + b & \text{if } \mathbf{y} \neq \emptyset, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Here $\theta = (\mathbf{w}, b)$, $\mathbf{w}$ is the weight vector and b is the bias term. From now on, for brevity, we use $f(\mathbf{X_y})$ instead of $f(\mathbf{X_y};\theta)$ to denote the score of segment $\mathbf{X_y}$.

To support early detection of events in time series data, we propose to use partial events as positive training examples (Fig. 2). In particular, we simulate the sequential arrival of training data as follows. Suppose the length of $\mathbf{X}^i$ is l^i. For each time $t = 1, \cdots, l^i$, let $\mathbf{y}_t^i$ be the part of event $\mathbf{y}^i$ that has already happened, i.e., $\mathbf{y}_t^i = \mathbf{y}^i \cap [1, t]$, which is possibly empty. Ideally, we want the output of the detector on time series $\mathbf{X}^i$ at time t to be the partial event, i.e.,

$$g(\mathbf{X}^i_{[1,t]}) = \mathbf{y}_t^i. \tag{3}$$

Note that $g(\mathbf{X}^i_{[1,t]})$ is not the output of the detector running on the entire time series $\mathbf{X}^i$. It is the output of the detector on the subsequence of time series $\mathbf{X}^i$ from the first frame to the tth frame only, i.e.,

$$g(\mathbf{X}^i_{[1,t]}) = \underset{\mathbf{y}\in\mathcal{Y}(t)}{\operatorname{argmax}}\, f(\mathbf{X}^i_{\mathbf{y}}). \tag{4}$$

From (3)–(4), the desired property of the score function is:

$$f(\mathbf{X}^i_{\mathbf{y}_t^i}) \geq f(\mathbf{X}^i_{\mathbf{y}})\ \forall \mathbf{y} \in \mathcal{Y}(t). \tag{5}$$

This constraint requires the score of the partial event $\mathbf{y}_t^i$ to be higher than the score of any other time series segment $\mathbf{y}$ that has been seen in the past, $\mathbf{y} \subset [1, t]$. This is illustrated in Fig. 3. Note that the score of the partial event is not required to be higher than the score of a future segment.

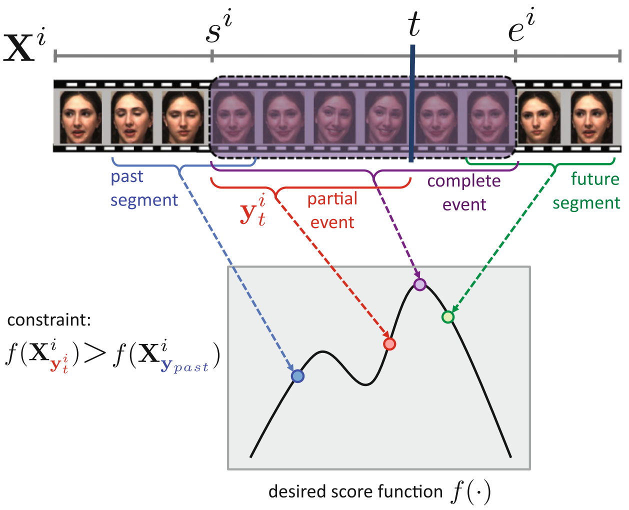

Fig. 3 The desired score function for early event detection: the complete event must have the highest detection score, and the detection score of a partial event must be higher than that of any segment that ends before the partial event. To learn this function, we explicitly consider partial events during training. At time t, the score of the truncated event (*red* segment) is required to be higher than the score of any segment in the past (e.g., *blue* segment); however, it is not required to be higher than the score of any future segment (e.g., *green* segment). This figure is best seen in color (Color figure online)

As in the case of SOSVM, the previous constraint can be required to be well satisfied by an adaptive margin. This margin is $\Delta(\mathbf{y}_t^i, \mathbf{y})$, the loss of the detector for outputting $\mathbf{y}$ when the desired output is $\mathbf{y}_t^i$ (in our case $\Delta(\mathbf{y}_t^i, \mathbf{y}) = 1 - \frac{2|\mathbf{y}_t^i \cap \mathbf{y}|}{|\mathbf{y}_t^i| + |\mathbf{y}|}$). The desired constraint is:

$$f(\mathbf{X}^i_{\mathbf{y}_t^i}) \geq f(\mathbf{X}^i_{\mathbf{y}}) + \Delta(\mathbf{y}_t^i, \mathbf{y})\ \forall \mathbf{y} \in \mathcal{Y}(t). \tag{6}$$

This constraint should be enforced for all $t = 1, \cdots, l^i$. As in the formulations of SVM and SOSVM, constraints are allowed to be violated by introducing slack variables, and we obtain the following learning formulation:

$$\underset{\mathbf{w},b,\xi^i\geq 0}{\text{minimize}}\ \frac{1}{2}||\mathbf{w}||^2 + \frac{C}{n}\sum_{i=1}^{n}\xi^i, \tag{7}$$

$$\text{s.t. } f(\mathbf{X}^i_{\mathbf{y}_t^i}) \geq f(\mathbf{X}^i_{\mathbf{y}}) + \Delta(\mathbf{y}_t^i, \mathbf{y}) - \frac{\xi^i}{\mu\left(\frac{|\mathbf{y}_t^i|}{|\mathbf{y}^i|}\right)}$$

$$\forall i, \forall t = 1, \cdots, l^i, \forall \mathbf{y} \in \mathcal{Y}(t). \tag{8}$$

Here $|\cdot|$ denotes the length function, and $\mu\left(\frac{|\mathbf{y}_t^i|}{|\mathbf{y}^i|}\right)$ is a function of the proportion of the event that has occurred at time t. $\mu\left(\frac{|\mathbf{y}_t^i|}{|\mathbf{y}^i|}\right)$ is a slack variable rescaling factor and should correlate with the importance of correctly detecting at time t whether the event $\mathbf{y}^i$ has happened. $\mu(\cdot)$ can be any arbitrary non-negative function, and in general, it should be a non-decreasing function in (0, 1]. In our experiments, we found the following piece-wise linear function a reasonable choice: $\mu(x) = 0$ for $0 < x \leq \alpha$; $\mu(x) = (x - \alpha)/(\beta - \alpha)$ for $\alpha < x \leq \beta$; and $\mu(x) = 1$ for $\beta < x \leq 1$ and $x = 0$. Here, α and β are tunable parameters. $\mu(0) = \mu(1)$ emphasizes that true rejection is as important as true detection of the complete event. This function is depicted in Fig. 5.

This learning formulation is an extension of SOSVM. From this formulation, we obtain SOSVM by not simulating the sequential arrival of training data, i.e., to set $t = l^i$ instead of $t = 1, \cdots, l^i$ in Constraint (8). Notably, our method does more than augment the set of training examples; it enforces monotonicity of the detector function, as shown in Fig. 4.

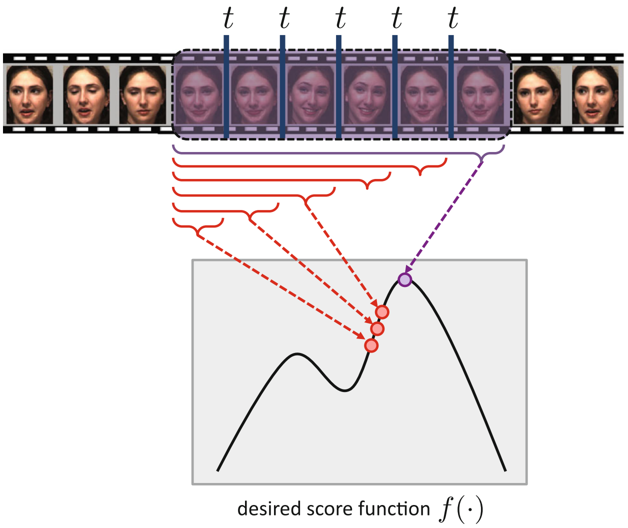

Fig. 4 Monotonicity requirement—the detection score of a partial event cannot exceed the score of an encompassing partial event. MMED provides a principled mechanism to achieve this monotonicity, which cannot be assured by a naive solution that simply augments the set of training examples

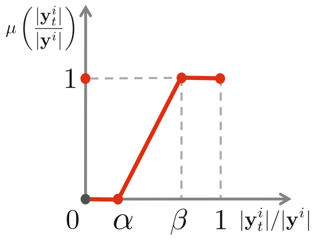

Fig. 5 μ—a function to weigh the importance of partially observed events. Here 0 and 1 correspond to the total absence and full completion of the event of interest, respectively. $\mu(0) = \mu(1)$ emphasizes that true rejection is as important as true detection of the complete event

For a better understanding of Constraint (8), let us analyze the constraint without the slack variable term and break it into three cases: (i) $t < s^i$ (event has not started); (ii) $t \geq s^i$, $\mathbf{y} = \emptyset$ (event has started; compare the partial event against the detection threshold); (iii) $t \geq s^i, \mathbf{y} \neq \emptyset$ (event has started; compare the partial event against any non-empty segment). Recall $f(\mathbf{X}_\emptyset) = 0$ and $\mathbf{y}^i_t = \emptyset$ for $t < s^i$, cases (i), (ii), (iii) lead to Constraints (9)–(11), respectively:

$$f(\mathbf{X}^i_{\mathbf{y}}) \leq -1 \;\forall \mathbf{y} \in \mathcal{Y}(s^i - 1)\backslash\{\emptyset\}, \tag{9}$$

$$f(\mathbf{X}^i_{\mathbf{y}^i_t}) \geq 1 \;\forall t \geq s^i, \tag{10}$$

$$f(\mathbf{X}^i_{\mathbf{y}^i_t}) \geq f(\mathbf{X}^i_{\mathbf{y}}) + \Delta(\mathbf{y}^i_t, \mathbf{y}) \;\forall t \geq s^i, \mathbf{y} \in \mathcal{Y}(t)\backslash\{\emptyset\}. \tag{11}$$

Constraint (9) prevents false detection when the event has not started. Constraint (10) requires successful recognition of partial events. Constraint (11) trains the detector to accurately localize the temporal extent of the partial events.

In the formulation presented above, we learn a detection score function that is a linear combination of the features in a feature vector, Eq. 2. However, this is not restricted to linearity because the feature function $\varphi(\cdot)$ can be non-linear. In particular, we can kernelize our algorithm by using explicit feature maps, which exist for several common kernels such as intersection (Maji and Berg 2009), $\mathcal{X}^2$ (Vedaldi and Zisserman 2010), and RBF (Le et al. 2013).

3.2 Optimization Algorithm

The proposed learning formulation Eq. (7) is convex, but it contains a large number of constraints. Following Tsochantaridis et al. (2005), we propose to use constraint generation to optimize it, i.e., we maintain a smaller subset of constraints and iteratively update it by adding the most violated ones. Constraint generation is guaranteed to converge to the global minimum. In our experiments described in Sect. 4, this usually converges within 20 iterations. Each iteration requires minimizing a convex quadratic objective. This objective is optimized using Cplex[1] in our implementation[2].

Each iteration of constraint generation requires finding the most violated constraints, one for each training time series. The most violated constraint for training time series $\mathbf{X}^i$ can be found by solving:

$$\hat{t}, \hat{\mathbf{y}} = \operatorname*{argmax}_{t, \mathbf{y} \in \mathcal{Y}(t)} \left(f(\mathbf{X}^i_{\mathbf{y}}) + \Delta(\mathbf{y}^i_t, \mathbf{y}) - f(\mathbf{X}^i_{\mathbf{y}^i_t}) \right) \mu\left(\frac{|\mathbf{y}^i_t|}{|\mathbf{y}^i|}\right).$$

Recall that $f(\mathbf{X}^i_{\mathbf{y}}) = \mathbf{w}^T \varphi(\mathbf{X}^i_{\mathbf{y}}) + b$ for $\mathbf{y} \neq \emptyset$, with $\mathbf{w}$ is the SVM weight vector at the current iteration. This optimization problem can be solved using exhaustive search, which has complexity $O((l_{max} - l_{min}){l^i}^2)$.

3.3 Loss Function and Empirical Risk Minimization

In Sect. 3.1, we have proposed a formulation for training early event detectors. This section provides further discussion on what exactly is being optimized. First, we briefly review the loss of SOSVM and its surrogate empirical risk. We then describe two general approaches for quantifying the loss of a detector on sequential data. In both cases, what Eq. (7) minimizes is an upper bound on the loss.

As previously explained, $\Delta(\mathbf{y}, \hat{\mathbf{y}})$ is the function that quantifies the loss associated with a prediction $\hat{\mathbf{y}}$, if the true output value is $\mathbf{y}$. Thus, in the setting of offline detection, the loss of a detector $g(\cdot)$ on a sequence-event pair $(\mathbf{X}, \mathbf{y})$ is quantified as $\Delta(\mathbf{y}, g(\mathbf{X}))$. Suppose the sequence-event pairs $(\mathbf{X}, \mathbf{y})$ are generated according to some distribution $P(\mathbf{X}, \mathbf{y})$, the loss

[1] www-01.ibm.com/software/integration/optimization/cplex-optimizer/.

[2] http://www.robots.ox.ac.uk/~minhhoai/projects/mmed.html.

of the detector g is

$$\mathcal{R}^{\Delta}_{true}(g) = \int_{\mathcal{X}\times\mathcal{Y}} \Delta(\mathbf{y}, g(\mathbf{X}))dP(\mathbf{X}, \mathbf{y}). \tag{12}$$

However, P is unknown so the performance of $g(.)$ is described by the empirical risk on the training data $\{(\mathbf{X}^i, \mathbf{y}^i)\}$, assuming they are generated i.i.d according to P. The empirical risk is

$$\mathcal{R}^{\Delta}_{emp}(g) = \frac{1}{n}\sum_{i=1}^{n} \Delta(\mathbf{y}^i, g(\mathbf{X}^i)). \tag{13}$$

It has been shown that SOSVM minimizes an upper bound on the empirical risk $\mathcal{R}^{\Delta}_{emp}$ (Tsochantaridis et al. 2005).

Due to the nature of continual evaluation, quantifying the loss of an online detector on streaming data requires aggregating the losses evaluated throughout the course of the data sequence. Let us consider the loss associated with a prediction $\mathbf{y} = g(\mathbf{X}^i_{[1,t]})$ for time series $\mathbf{X}^i$ at time t as $\Delta(\mathbf{y}^i_t, \mathbf{y})\mu\left(\frac{|\mathbf{y}^i_t|}{|\mathbf{y}^i|}\right)$. Here $\Delta(\mathbf{y}^i_t, \mathbf{y})$ accounts for the difference between the output $\mathbf{y}$ and true truncated event $\mathbf{y}^i_t$. $\mu\left(\frac{|\mathbf{y}^i_t|}{|\mathbf{y}^i|}\right)$ is the scaling factor; it depends on how much the temporal event $\mathbf{y}^i$ has happened. Two possible ways for aggregating these loss quantities is to use their maximum or average. They lead to two different empirical risks for a set of training time series:

$$\mathcal{R}^{\Delta,\mu}_{max}(g) = \frac{1}{n}\sum_{i=1}^{n} \max_t \left\{\Delta(\mathbf{y}^i_t, g(\mathbf{X}^i_{[1,t]}))\mu\left(\frac{|\mathbf{y}^i_t|}{|\mathbf{y}^i|}\right)\right\},$$

$$\mathcal{R}^{\Delta,\mu}_{mean}(g) = \frac{1}{n}\sum_{i=1}^{n} \operatorname*{mean}_t \left\{\Delta(\mathbf{y}^i_t, g(\mathbf{X}^i_{[1,t]}))\mu\left(\frac{|\mathbf{y}^i_t|}{|\mathbf{y}^i|}\right)\right\}.$$

In the following, we state and prove a proposition that establishes that the learning formulation given in Eq. 7 minimizes an upper bound of the above two empirical risks.

Proposition *Denote by* $\boldsymbol{\xi}^*(g)$ *the optimal solution of the slack variables in Eq. (7) for a given detector* g*, then* $\frac{1}{n}\sum_{i=1}^{n}\xi^{i*}$ *is an upper bound on the empirical risks* $\mathcal{R}^{\Delta,\mu}_{max}(g)$ *and* $\mathcal{R}^{\Delta,\mu}_{mean}(g)$.

Proof Consider Constraint (8) with $\mathbf{y} = g(\mathbf{X}^i_{[1,t]})$ and together with the fact that $f(\mathbf{X}^i_{g(\mathbf{X}^i_{[1,t]})}) \geq f(\mathbf{X}^i_{\mathbf{y}^i_t})$, we have

$$\xi^{i*} \geq \Delta(\mathbf{y}^i_t, g(\mathbf{X}^i_{[1,t]}))\mu\left(\frac{|\mathbf{y}^i_t|}{|\mathbf{y}^i|}\right) \;\forall t. \tag{14}$$

$$\Rightarrow \xi^{i*} \geq \max_t \left\{\Delta(\mathbf{y}^i_t, g(\mathbf{X}^i_{[1,t]}))\mu\left(\frac{|\mathbf{y}^i_t|}{|\mathbf{y}^i|}\right)\right\}. \tag{15}$$

$$\Rightarrow \frac{1}{n}\sum_{i=1}^{n}\xi^{i*} \geq \mathcal{R}^{\Delta,\mu}_{max}(g) \geq \mathcal{R}^{\Delta,\mu}_{mean}(g). \tag{16}$$

This completes the proof of the proposition. This proposition justifies the objective of the learning formulation. □

3.4 Slack Rescaling Versus Margin Rescaling: A Comparison

This section describes an alternative formulation to Eq. 7 and its disadvantages. Recall in Eq. 7, we use $\mu\left(\frac{|\mathbf{y}^i_t|}{|\mathbf{y}^i|}\right)$ to rescale the slack variable ξ^i to weigh the importance for detecting the partial event at time t. Alternatively, one can rescale the the margin $\Delta(\mathbf{y}^i_t, \mathbf{y})$, which leads to the following formulation:

$$\operatorname*{minimize}_{\mathbf{w},b,\xi^i\geq 0} \frac{1}{2}||\mathbf{w}||^2 + \frac{C}{n}\sum_{i=1}^{n}\xi^i, \tag{17}$$

$$\text{s.t. } f(\mathbf{X}^i_{\mathbf{y}^i_t}) \geq f(\mathbf{X}^i_{\mathbf{y}}) + \Delta(\mathbf{y}^i_t, \mathbf{y})\mu\left(\frac{|\mathbf{y}^i_t|}{|\mathbf{y}^i|}\right) - \xi^i$$
$$\forall i, \forall t = 1, \cdots, l^i, \forall \mathbf{y} \in \mathcal{Y}(t). \tag{18}$$

While it is possible to use the above formulation for early event detection, it has a disadvantage compared with the formulation proposed in Eq. 7. To see this disadvantage, consider the difference between these two formulations, which lies at their constraints, Constraint (8) versus (18). Consider these two constraints for a particular time series $\mathbf{X}^i$ and at a particular time t. Both constraints adjust the original constraint, $f(\mathbf{X}^i_{\mathbf{y}^i_t}) \geq f(\mathbf{X}^i_{\mathbf{y}}) + \Delta(\mathbf{y}^i_t, \mathbf{y})$, based on the importance for recognizing the partial event at time t. The former reweighs the original constraint, while the latter reweighs the margin. In reality, not every event can be detected as soon as a small fraction of the event occurs; therefore, it is important to reweigh the constraint and even to deactivate it. This can be achieved using the former constraint, but not the latter. For example, the former allows the deactivation of itself by setting the scaling factor $\mu\left(\frac{|\mathbf{y}^i_t|}{|\mathbf{y}^i|}\right)$ to 0 (for $|\mathbf{y}^t_t|/|\mathbf{y}^i| \leq \alpha$ as in Fig. 5), while the latter does not.

3.5 Event Detectors in Action

The main novelty of this paper is the learning formulation to train a detector for early detection. Once the detector has been trained, it can be used in several ways, depending on the event type and the application. This section discusses one possible approach to use the trained detector at test time.

Early event detection requires realtime processing, and therefore, target events, if they occur more than once, must be detected sequentially. We propose a detection mechanism as follows. The detector reads from a stream of data and keeps a sequence of observations in its memory. It continuously monitors for the occurrence of a target event. If a target event is detected, the temporal extent of the event is returned. If a target event is recognized as complete, the detector's memory is cleared and the process restarts to detect the upcoming target event. Thus, at any time, the detector needs to detect at most one target event. Let t_0 be the beginning of the data stream in consideration, and suppose the length of the partial and full events that we need to detect are bounded by l_{min} and l_{max}. We denote $\mathcal{Y}(t_0, t)$ to be the set of length-bounded time intervals from the time t_0 to time t:

$$\mathcal{Y}(t_0, t) = \{\mathbf{y} \in \mathbb{N}^2 | \mathbf{y} \subset [t_0, t], l_{min} \le |\mathbf{y}| \le l_{max}\} \cup \{\emptyset\}.$$

The output of the detector at time t is given by:

$$g(\mathbf{X}_{[t_0,t]}) = \underset{\mathbf{y} \in \mathcal{Y}(t_0,t)}{\operatorname{argmax}} f(\mathbf{X}_\mathbf{y}). \tag{19}$$

The computation for finding the detector's output at time t can be reused at time $t+1$ because:

$$\max_{\mathbf{y} \in \mathcal{Y}(t_0,t+1)} f(\mathbf{X}_\mathbf{y}) = \max\left\{\max_{\mathbf{y} \in \mathcal{Y}(t_0,t)} f(\mathbf{X}_\mathbf{y}), f(\mathbf{X}_{\hat{\mathbf{y}}_{t+1}})\right\}.$$

where $\hat{\mathbf{y}}_{t+1}$ is the segment that attains the maximum detection score among all segments that terminate at $t+1$:

$$\hat{\mathbf{y}}_{t+1} = \underset{\mathbf{y} \in \mathcal{Y}(t_0,t+1), \mathbf{y}(2)=t+1}{\operatorname{argmax}} f(\mathbf{X}_\mathbf{y}). \tag{20}$$

To compute the detector's output at time $t+1$, it is sufficient to compare the output of the detector at time t and the events that terminate at $t+1$. The complexity of this procedure is $\mathbf{O}(l_{max} - l_{min})$.

Using the above detection mechanism for detecting multiple target events in sequential data requires knowing when a target event has completed (so t_0 can be reset to a new value). The completion of a target event can be determined by monitoring the current best event, $\hat{\mathbf{y}}_t$. We consider a detected event has completed if the value of $f(\mathbf{X}_{\hat{\mathbf{y}}_t})$ falls below some threshold.

To detect target events of multiple classes, in this paper, we train and use the event detectors separately, one for each target event class. Another approach is to jointly train multiple event detectors and jointly detect multiple events, as in the case of joint segmentation and recognition (Crammer and Singer 2001; Hoai et al. 2011). This will be explored in our future work.

4 Experiments

This section describes our experiments on several publicly available datasets of varying complexity.

4.1 Evaluation Criteria

This section describes several criteria for evaluating the accuracy and timeliness of detectors. We used the area under the Receiver Operating Characteristic (ROC) curve for accuracy comparison, normalized time to detection (NTtoD) for benchmarking the timeliness of detection, and $F1$-score for evaluating localization quality.

4.1.1 Area under the ROC Curve

Consider testing a detector on a set of time series. The FPR of the detector is defined as the fraction of time series that the detector fires before the event of interest starts. The True Positive Rate (TPR) is defined as the fraction of time series that the detector fires during the event of interest. A detector typically has a detection threshold that can be adjusted to trade off high TPR for low FPR and vice versa. By varying this detection threshold, we can generate a ROC curve, which is the function of TPR against FPR. We use the area under the ROC for evaluating the detector accuracy.

4.1.2 AMOC Curve

To evaluate the timeliness of detection we used NTtoD which is defined as follows. Given a testing time series where the event of interest occurs from s to e. Suppose the detector starts to fire at time t. For a successful detection, $s \le t \le e$, we define the NTtoD as the fraction of event that has occurred, i.e., $\frac{t-s+1}{e-s+1}$. NTtoD is defined as 0 for a false detection ($t < s$) and ∞ for a false rejection ($t > e$). By adjusting the detection threshold, one can achieve lower NTtoD at the cost of higher FPR and vice versa. For a complete characteristic picture, we varied the detection threshold and plotted the curve of NToD versus FPR. This is referred as the Activity Monitoring Operating Curve (AMOC) (Fawcett and Provost 1999).

4.1.3 F1-Score curve

The ROC and AMOC curves, however, do not provide a measure for how well the detector can localize the event of interest. For this purpose, we propose to use the frame-based $F1$-scores. Consider running a detector on a times series. At time t the detector output the segment $\mathbf{y}$ while the ground truth (possibly) truncated event is $\mathbf{y}^*$. The $F1$-score is defined as the harmonic mean of the precision and recall values: $F1 := \frac{2 \times Precision \times Recall}{Precision + Recall}$, with $Precision := \frac{|\mathbf{y} \cap \mathbf{y}^*|}{|\mathbf{y}|}$ and $Recall := \frac{|\mathbf{y} \cap \mathbf{y}^*|}{|\mathbf{y}^*|}$. For a new test time series, we can simulate the sequential arrival of data and record the $F1$-scores as the event of interest unrolls from 0 to 100 %. We refer to this as the *F1-score curve*.

4.2 Synthetic Data

We first validated the performance of MMED on a synthetically generated dataset of 200 time series. Each time series contained one instance of the event of interest, signal Fig. 6a,i, and several instances of other events, signals Fig. 6a,ii–iv. Some examples of these time series are shown in Fig. 6b. We randomly split the data into training and testing subsets of equal sizes. During testing we simulated

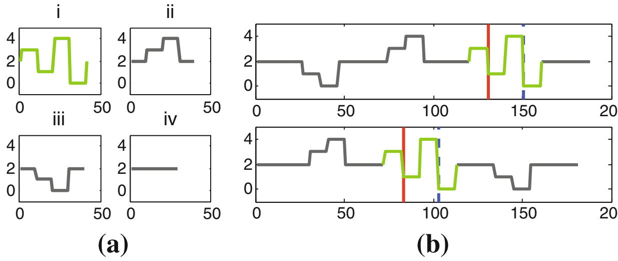

Fig. 6 Synthetic data experiment. (**a**) Time series were created by concatenating the event of interest (*i*) and several instances of other events (*ii*)–(*iv*). (**b**) Examples of testing time series; the *solid vertical red lines* mark the moments that our method starts to detect the event of interest while the *dashed blue lines* are the results of SOSVM (Color figure online)

the sequential arrival of data and recorded the moment that MMED started to detect the start of the event of interest. With 100 % precision, MMED detected the event when it had completed 27.5 % of the event. For comparison, SOSVM required observing 77.5 % of the event for a positive detection. Examples of testing time series and results are depicted in Fig. 6b. The events of interest are drawn in green and the solid vertical red lines mark the moments that our method started to detect these events. The dashed vertical blue lines are the results of SOSVM. Notably, this result reveals an interesting capability of MMED. For the time series in this experiment, the change in signal values from 3 to 1 is exclusive to the target events. MMED was trained to recognize partial events, it implicitly discovered this unique behavior, and it detected the target events as soon as this behavior occurred. In this experiment, we represented each time series segment by the L_2-normalized histogram of signal values in the segment (normalized to have unit norm). We used linear SVM with $C = 1,000, \alpha = 0, \beta = 1$.

4.3 Auslan Dataset—Australian Sign Language

This section describes our experiments on a publicly available dataset (Kadous 2002) that contains 95 Auslan signs, each with 27 examples. The signs were captured from a native signer using position trackers and instrumented gloves. The location of two hands, the orientation of the palms, and the bending of the fingers were recorded. We considered detecting the sentence "I love you" in monologues obtained by concatenating multiple signs. In particular, each monologue contained an I-love-you sentence which was preceded and succeeded by 15 random signs. The I-love-you sentence was an ordered concatenation of random samples of three signs: "I", "love", and "you". We created 100 training and 200 testing monologues from disjoint sets of sign samples; the first 15 examples of each sign were used to create training monologues while the last 12 examples were used for testing monologues. The average lengths and standard deviations of the monologues and the I-love-you sentences were $1,836 \pm 38$ and 158 ± 6 respectively.

Previous work (Kadous 2002) reported high recognition performance on this dataset using HMMs. Following their success, we implemented a continuous density HMM for I-love-you sentences. Our HMM implementation consisted of 10 states, and each was a mixture of 4 Gaussians. To use the HMM for detection, we adopted a sliding window approach; the window size was fixed to the average length of the I-love-you sentences.

Inspired by the high recognition rate of HMM, we constructed the feature representation for SVM-based detectors (SOSVM and MMED) as follows. We first trained a Gaussian mixture model of 20 Gaussians for the frames extracted from the I-love-you sentences. Each frame was then associated with a 20×1 log-likelihood vector. We retained the top three values of this vector, zeroing out the other values, to create a frame-level feature representation. This is often referred to as a soft quantization approach. To compute the feature vector for a given window, we divided the window into two roughly equal halves, the mean feature vector of each half was calculated, and the concatenation of these mean vectors was used as the feature representation of the window.

A naive strategy for early detection is to use truncated events as positive examples. For comparison, we implemented *Seg-[0.5,1]*, a binary SVM that used the first halves of the I-love-you sentences in addition to the full sentences as positive training examples. Negative training examples were random segments that had no overlapping with the I-love-you sentences.

We repeated our experiment 10 times and recorded the average performance. Regarding the detection accuracy, all methods except SVM-[0.5,1] performed similarly well. The ROC areas for HMM, SVM-[0.5,1], SOSVM, and MMED were 0.97, 0.92, 0.99, and 0.99, respectively. However, when comparing the timeliness of detection, MMED outperformed the others by a large margin. For example, at 10 % FPR, our method detected the I-love-you sentence when it observed the first 37 % of the sentence. At the same FPR, the best alternative method required seeing 62 % of the sentence. The full AMOC curves are depicted in Fig. 7. In this experiment, we used linear SVM with $C = 1, \alpha = 0.25, \beta = 1$.

4.4 Extended Cohn–Kanade Dataset—Facial Expression

The extended Cohn–Kanade dataset (CK+) (Lucey et al. 2010) contains 327 facial image sequences from 123 subjects performing one of seven discrete emotions: anger, contempt, disgust, fear, happiness, sadness, and surprise. Each of the sequences contains images from onset (neutral frame) to peak expression (last frame). We considered the task of detecting negative emotions: anger, disgust, fear, and sadness.

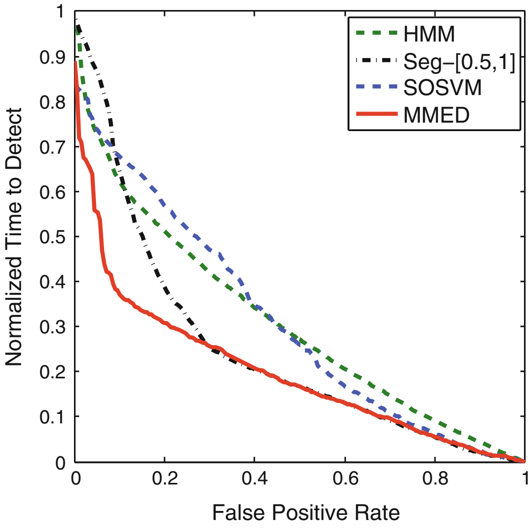

Fig. 7 AMOC curves on Auslan dataset; at the same false positive rate, MMED detects the event of interest sooner than the others. This figure is best seen in color (Color figure online)

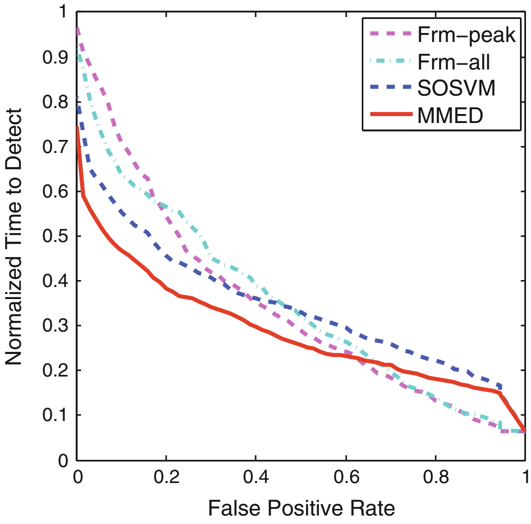

Fig. 8 AMOC curves on CK+ dataset; at the same false positive rate, MMED detects the event of interest sooner than the others. This figure is best seen in color (Color figure online)

We used the same representation as Lucey et al. (2010), where each frame is represented by the canonical normalized appearance feature, referred to as CAPP in Lucey et al. (2010). For comparison purposes, we implemented two frame-based SVMs: *Frm-peak* was trained on peak frames of the training sequences while *Frm-all* was trained using all frames between the onset and offset of the facial action. Frame-based SVMs can be used for detection by classifying individual frames. In contrast, SOSVM and MMED are segment-based. Since a facial expression is a deviation of the neutral expression, we represented each segment of an emotion sequence by the difference between the end frame and the start frame. Even though the start frame was not necessarily a neutral face, this representation led to good recognition results.

We randomly divided the data into disjoint training and testing subsets. The training set contained 200 sequences with equal numbers of positive and negative examples. For reliable results, we repeated our experiment 20 times and recorded the average performance. Regarding the detection accuracy, segment-based SVMs outperformed frame-based SVMs. The ROC areas (mean and standard deviation) for Frm-peak, Frm-all, SOSVM, MMED are 0.82 ± 0.02, 0.84 ± 0.03, 0.96 ± 0.01, and 0.97 ± 0.01, respectively. Comparing the timeliness of detection, our method was significantly better than the others, especially at low FPR. For example, at 10 % FPR, Frm-peak, Frm-all, SOSVM, and MMED can detect the expression when it completes 71, 64, 55, and 47 % respectively. Figure 8 plots the AMOC curves, and Fig. 9 displays some qualitative results. In this experiment, we used a linear SVM with $C = 1000$, $\alpha = 0$, $\beta = 0.5$.

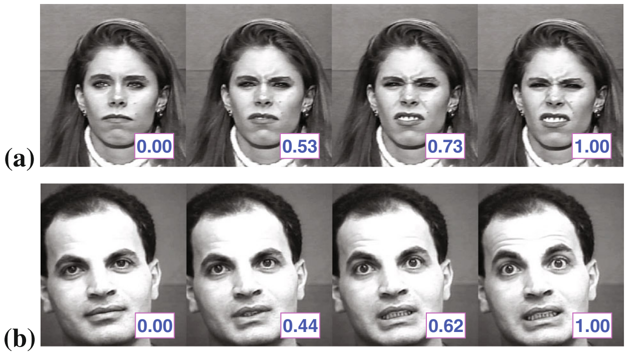

Fig. 9 Disgust (**a**) and fear (**b**) detection on CK+ dataset. From *left* to *right*: the onset frame, the frame at which MMED fires, the frame at which SOSVM fires, and the peak frame. The number in each image is the corresponding NTtoD

4.5 Weizmann Dataset—Human Action

The Weizmann dataset contains 90 video sequences of 9 people, each performing 10 actions. Each video sequence in this dataset only consists of a single action. To measure the accuracy and timeliness of detection, we performed experiments on longer video sequences that were created by concatenating existing single-action sequences. Following Gorelick et al. (2007), we extracted binary masks and computed a Euclidean distance transform for frame-level features. Frame-level feature vectors were clustered using k-means to create a codebook of 100 temporal words. Subsequently, each frame was represented by the ID of the corresponding codebook entry and each segment of a time series was represented by the histogram of temporal words associated with frames inside the segment.

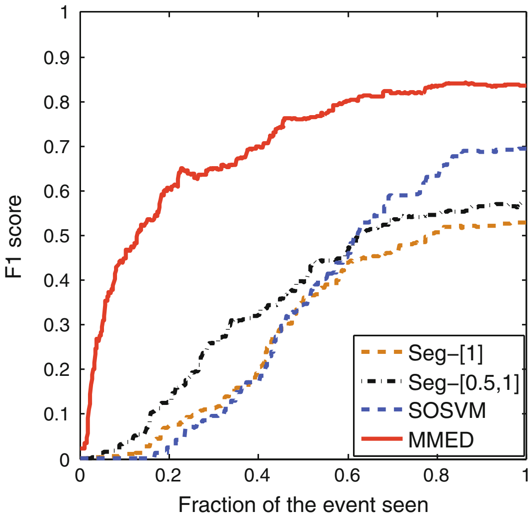

Fig. 10 F1-score curves on Weizmann dataset; MMED provides better localization for the event of interest, especially when the fraction of the event observed is small. This figure is best seen in color (Color figure online)

We trained a detector for each action class, but considered them one by one. We created 9 long video sequences, each composed of 10 videos of the same person and had the event of interest at the end of the sequence. We performed leave-one-out cross validation; each cross validation fold trained the event detector on 8 sequences and tested it on the leave-out sequence. For the testing sequence, we computed the normalized time to detection at 0 % false positive rate. This FPR was achieved by raising the threshold for detection so that the detector would not fire before the event started. We calculated the median normalized time to detection across 9 cross validation folds and averaged these median values across 10 action classes; the resulting values for Seg-[1], Seg-[0.5,1], SOSVM, MMED are 0.16, 0.23, 0.16, and 0.10 respectively. Here Seg-[1] was a segment-based SVM, trained to classify the segments corresponding to the complete action of interest. Seg-[0.5,1] was similar to Seg-[1], but used the first halves of the action of interest as additional positive examples. For each testing sequence, we also generated a F1-score curve as described in Sect. 4.1. Figure 10 displays the $F1$-score curves of all methods, averaged across different actions and different cross-validation folds. MMED significantly outperformed the other methods. The superiority of MMED over SOSVM was especially large when the fraction of the event observed was small. This was because MMED was trained to detect truncated events while SOSVM was not. Though also trained with truncated events, Seg-[0.5,1] performed relatively poor because it was not optimized to produce correct temporal extent of the event. In this experiment, we used the linear SVM with $C = 1000, \alpha = 0, \beta = 1$.

5 Conclusions

This paper addressed the problem of early event detection. We proposed MMED, a temporal classifier specialized in detecting events as soon as possible. Moreover, MMED provides localization for the temporal extent of the event. MMED is based on SOSVM, but extends it to anticipate sequential data. During training, we simulate the sequential arrival of data and train a detector to recognize incomplete events. It is important to emphasize that we train a *single* event detector to recognize *all* partial events and that our method does more than augment the set of training examples. Our method is particularly suitable for events that cannot be reliably detected by classifying individual frames; detecting this type of events requires pooling information from a supporting window. Experiments on datasets of varying complexity, from synthetic data and sign language to facial expression and human actions, showed that our method often made faster detections while maintaining comparable or even better accuracy. Furthermore, our method provided better localization for the target event, especially when the fraction of the seen event was small. In this paper, we illustrated the benefits of our approach in the context of human activity analysis, but our work can be applied to many other domains. The active training approach to detect partial temporal events can be generalized to detect truncated spatial objects (Vedaldi and Zisserman 2009).

Acknowledgments This work was supported by the National Science Foundation (NSF) under Grant No. RI-1116583. The authors would like to thank Yuan Shi for the useful discussion on early detection, Lorenzo Torresani for the suggestion of F1 curves, Maxim Makatchev for the discussion about AMOC curves, Tomas Simon for AU data, and Patrick Lucey for providing CAPP features for the Extended Cohn–Kanade dataset. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.

References

Ali, S., & Shah, M. (2010). Human action recognition in videos using kinematic features and multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(2), 288–303.

Amer, M. R., Xie, D., Zhao, M., Todorovic, S., & Zhu, S. C. (2012). Cost-sensitive top-down/bottom-up inference for multiscale activity recognition. In *Proceedings of the european conference on computer vision*.

Bobick, A. F., & Davis, J. W. (2001). The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *23*(3), 257–267.

Brand, M., Oliver, N., & Pentland, A. (1997). Coupled hidden Markov models for complex action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Brendel, W., & Todorovic, S. (2011). Learning spatiotemporal graphs of human activities. In *Proceedings of the international conference on computer vision*.

Brown, P. F., deSouza, P. V., Mercer, R. L., Pietra, V. J. D., & Lai, J. C. (1992). Class-based n-gram models of natural language. *Computational Linguistics*, *18*(4), 467–479.

Chomat, O., & Crowley, J. (1999). Probabilistic recognition of activity using local appearance. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Cohn, J., Simon, T., Matthews, I., Yang, Y., Nguyen, M. H., Tejera, M., Zhou, F., & De la Torre, F. (2009). Detecting depression from facial actions and vocal prosody. In *Proceedings of international conference on affective computing and intelligent interaction*.

Cooper, H., & Bowden, R. (2009). Learning signs from subtitles: A weakly supervised approach to sign language recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Crammer, K., & Singer, Y. (2001). On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, *2*, 265–292.

Davis, J., & Tyagi, A. (2006). Minimal-latency human action recognition using reliable-inference. *Image and Vision Computing*, *24*(5), 455–472.

Desobry, F., Davy, M., & Doncarli, C. (2005). An online kernel change detection algorithm. *IEEE Transaction on Signal Processing*, *53*(8), 2961–2974.

Dollár, P., Rabaud, V., Cottrell, G., & Belongie, S. (2005). Behavior recognition via sparse spatio-temporal features. In *ICCV Workshop on visual surveillance and performance evaluation of tracking and surveillance*.

Duchenne, O., Laptev, I., Sivic, J., Bach, F. R., & Ponce, J. (2009). Automatic annotation of human actions in video. In *Proceedings of the international conference on computer vision*.

Efros, A., Berg, A., Mori, G., & Malik, J. (2003). Recognizing action at a distance. In *Proceedings of the international conference on computer vision*.

Ellis, C., Masood, S., Tappen, M. F., LaViola, J. J., & Sukthankar, R. (2013). Exploring the trade-off between accuracy and observational latency in action recognition. *International Journal of Computer Vision*, *101*(3), 420–436.

Fawcett, T., & Provost, F. (1999). Activity monitoring: Noticing interesting changes in behavior. In *Proceedings of the SIGKDD conference on knowledge discovery and data mining*.

Gorelick, L., Blank, M., Shechtman, E., Irani, M., & Basri, R. (2007). Actions as space-time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(12), 2247–2253.

Haider, P., Brefeld, U., & Scheffer, T. (2007). Supervised clustering of streaming data for email batch detection. In *Proceedings of the international conference on machine learning*.

Hoai, M., & De la Torre, F. (2012a). Max-margin early event detectors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Hoai, M., & De la Torre, F. (2012b). Maximum margin temporal clustering. In *Proceedings of international conference on artificial intelligence and statistics*.

Hoai, M., Lan, Z. Z., & De la Torre, F. (2011). Joint segmentation and classification of human actions in video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Jhuang, H., Serre, T., Wolf, L., & Poggio, T. (2007). A biologically inspired system for action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Kadous, M. (2002). *Temporal classification: Extending the classification paradigm to multivariate time series*. PhD thesis, The University of New South Wales.

Ke, Y., Sukthankar, R., & Hebert, M. (2005). Efficient visual event detection using volumetric features. In *Proceedings of the international conference on computer vision*.

Kim, K. J. (2003). Financial time series forecasting using support vector machines. *Neurocomputing*, *55*(1–2), 307–319.

Klaser, A., Marszalek, M., Schmid, C., & Zisserman, A. (2010). Human focused action localization in video. In *Proceedings of international workshop on sign, gesture, activity*.

Lan, T., Wang, Y., & Mori, G. (2011). Discriminative figure-centric models for joint action localization and recognition. In *Proceedings of the international conference on computer vision*.

Le, Q. V., Sarlos, T., & Smola, A. (2013). Fastfood—approximating kernel expansions in loglinear time. In *Proceedings of the international conference on machine learning*.

Liu, J., Kuipers, B., & Savarese, S. (2011). Recognizing human actions by attributes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Lucey, P., Cohn, J. F., Kanade, T., Saragih, J., Ambadar, Z., & Matthews, I. (2010). The extended Cohn–Kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression. In *CVPR Workshop on human communicative behavior analysis*.

Maji, S., & Berg, A. C. (2009). Max-margin additive classifiers for detection. In *Proceedings of the international conference on computer vision*.

Marin-Jiménez, M. J., Zisserman, A., & Ferrari, V. (2011). "Here's looking at you, kid". Detecting people looking at each other in videos. In *Proceedings of the British machine vision conference*.

Masood, S., Ellis, C., Nagaraja, A., & Tappen, M. (2011). Measuring and reducing observational latency when recognizing actions. In *Proceedings of the international conference on computer vision*.

Mauthner, T., Roth, P., & Bischof, H. (2009). Action recognition from a small number of frames. In *Computer vision winter workshop*.

Nam, Y., Wohn, K., & Lee-Kwang, H. (1999). Modeling and recognition of hand gesture using colored petri nets. *IEEE Transactions on Systems, Man and Cybernetics*, *29*(5), 514–521.

Neill, D., Moore, A., & Cooper, G. (2006). A bayesian spatial scan statistic. In *Advances in neural information processing systems*.

Nguyen, M. H., Torresani, L., De la Torre, F., & Rother, C. (2009). Weakly supervised discriminative localization and classification: A joint learning process. In *Proceedings of the international conference on computer vision*.

Nguyen, M. H., Simon, T., De la Torre, F., & Cohn, J. (2010). Action unit detection with segment-based SVMs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Niebles, J. C., Chen, C. W., & Fei-Fei, L. (2010). Modeling temporal structure of decomposable motion segments for activity classification. In *Proceedings of the european conference on computer vision*.

Nowozin, S., & Shotton, J. (2012). Action points: A representation for low-latency online human action recognition. *Microsoft Research Technical Report MSR-TR-2012-68*, Cambridge.

Oh, S. M., Rehg, J. M., Balch, T., & Dellaert, F. (2008). Learning and inferring motion patterns using parametric segmental switching linear dynamic systems. *International Journal of Computer Vision*, *77*(1–3), 103–124.

Parameswaran, V., & Chellappa, R. (2006). View invariance for human action recognition. *International Journal of Computer Vision*, *66*(1), 83–101.

Patron-Perez, A., Marszalek, M., Zisserman, A., & Reid, I. (2010). High five: Recognising human interactions in TV shows. In *Proceedings of British machine vision conference*.

Pei, M., Jia, Y., & Zhu, S. C. (2011). Parsing video events with goal inference and intent prediction. In *Proceedings of the international conference on computer vision*.

Reddy, K. K., & Shah, M. (2012). Recognizing 50 human action categories of web videos. *Machine Vision and Applications*, *24*(5), 971–981.

Ryoo, M. (2011). Human activity prediction: Early recognition of ongoing activities from streaming videos. In *Proceedings of the international conference on computer vision*.

Ryoo, M. S., & Aggarwal, J. K. (2009). Semantic representation and recognition of continued and recursive human activities. *International Journal of Computer Vision*, *32*(1), 1–24.

Satkin, S., & Hebert, M. (2010). Modeling the temporal extent of actions. In *Proceedings of the european conference on computer vision*.

Schindler, K., & Van Gool, L. (2008). Action snippets: How many frames does human action recognition require? In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Shi, Y., Nguyen, M. H., Blitz, P., French, B., Fisk, S., De la Torre, F., Smailagic, A., & Siewiorek, D. (2010). Personalized stress detection from physiological measurements. In *International symposium on quality of life technology*.

Smith, P., da Vitoria Lobo, N., & Shah, M. (2005). Temporal boost for event recognition. In *Proceedings of the international conference on computer vision*.

Taskar, B., Guestrin, C., & Koller, D. (2003). Max-margin Markov networks. In *Advances in neural information processing systems*.

Tran, S. D., & Davis, L. S. (2008). Event modeling and recognition using Markov logic networks. In *Proceedings of the european conference on computer vision*.

Tsochantaridis, I., Joachims, T., Hofmann, T., & Altun, Y. (2005). Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, *6*, 1453–1484.

Vedaldi, A., & Zisserman, A. (2009). Structured output regression for detection with partial truncation. In *Advances in neural information processing systems*.

Vedaldi, A., & Zisserman, A. (2010). Efficient additive kernels via explicit feature maps. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Yacoob, Y., & Black, M. J. (1999). Parameterized modeling and recognition of activities. *Computer Vision and Image Understanding*, *73*(2), 232–247.

Yang, Y., & Shah, M. (2012). Complex events detection using data-driven concepts. In *Proceedings of the european conference on computer vision*.